

## Citation 2

(Translation of Relevant parts and Abstract)

Japanese Patent Application Laying Open (KOKAI) No. 10-207717

laid open to the public August 7, 1998

Japanese Patent Application No. 9-6007

filed January 17, 1997

Internal Priority(ies) claimed: None

Applicant(s): Matsushita Electric Industrial Co., Ltd., Osaka, Japan

Inventor(s): Toshimichi MATSUZAKI et al., Japanese citizens

Title of Invention: MICROCOMPUTER

Detailed Description of the Invention:

[0027]

Embodiments of the Invention

FIG. 1 is a block diagram showing main components of a microcomputer according to an embodiment of the present invention. The microcomputer according to the present invention includes a register file 1, a CPU 2, a context saving memory 3, a memory 4, and a transfer unit 5.

[0028]

The register file 1 consists of register set A (GRA) and register set B (GAB). Whenever task switching occurs, the CPU 2 switches between register sets A and B and uses one of them to execute a task of CPU 2.

[0029]

The CPU 2 executes each task in turn while it switches tasks. When a task switching event occurs, the CPU 2 switches from one task to another in ascending or descending order of task number according to a task enable flag of the transfer unit 5. Task switching events include a periodic timer interrupt, a serial transfer interrupt, and a task switching instruction, for example.

[0030]

The context saving memory 3 is a memory for saving information concerning a task that is not currently being executed by the CPU 2. Information concerning a task is context including the contents of a register set, the contents of a processor status word (PSW), the contents of a program counter (PC) and so on.

[0031]

The memory 4 stores data to be processed by the CPU 2 during execution of a task. The transfer unit 5 restores the context of the next task to be executed by the CPU 2 from the context saving memory 3 to a register set that is not in use and switches from one register set to another whenever a task switching event occurs. While the CPU 2 is executing the next task by using the new register set, the transfer unit 5 saves the contents of the register set used before the switching into the context saving memory 3.

[0062]

FIG. 5 is a flowchart of the processing at the time of the task switching. In the drawing, at the time of the task switching the "CPU, transfer unit" and the "transfer unit" show a flow in which the CPU 2 and the transfer unit 5 are synchronized and processed in parallel, and a flow which is independently processed by the transfer unit 5, respectively.

[0063]

The flowchart of the drawing shows a flow in which, while the CPU 2 is in the midst of executing a task 1 by using the register set A, a task switching request occurs, and the task is switched to a task 2. When the task switching request occurs, the CPU 2 temporarily stops the task in the midst of being executed, and moves to a task switching processing. In this task switching, the CPU 2 is processed by a PLA 209 similarly to an interruption processing, and the transfer unit 5 is processed by a sequencer 55.

Explanation of the Figures:

FIG. 1

1 Register file

- 3 Context saving memory
- 4 Memory
- 5 Transfer unit

FIG. 4

- 1 Register file
- 51 Transfer control unit
- 54 RAM address controller
- 55 Sequencer
- 59 Task enable flag controller
- 62 Task number controller
- 64 Memory control unit
- 65 CTXMEM write controller
- 66 CTXMEM read controller
- 67 Register write controller
- 68 Register read controller

FIG. 5

- #1 CPU, transfer unit
- #2 Task switching request
- #3 Task 1 (GRA)
- #4 Return OAB save step 51
- #5 Return PC save step 52
- #6 Return PSW save step 53
- #7 Register return of task 2 step 54
- Return register (GRB)
- #8 PC return of task 2 step 55
- #9 PSW return of task 2 step 56
- #10 Timer set of task 2 step 57
- #11 OAB return of task 2 step 58
- #12 Register set switching step 59
- #13 Save operation activation of register set GRA step 60
- #14 Branch to task 2 step 61
- #15 Task 2 (GRB)
- #16 Transfer unit
- #17 Save start of register set (GRA)
- #18 Register save of task 1 step 62
- Save register (GRA)
- #19 Save completion of register set (GRA)

**PATENT ABSTRACTS OF JAPAN**

(11)Publication number : 10-207717  
(43)Date of publication of application : 07.08.1998

---

(51)Int.CI. G06F 9/46  
G06F 9/46

---

(21)Application number : 09-006007 (71)Applicant : MATSUSHITA  
ELECTRIC IND CO  
LTD  
(22)Date of filing : 17.01.1997 (72)Inventor : MATSUZAKI  
TOSHIMICHI  
TANAKA KEISUKE

---

**(54) MICROCOMPUTER**

**(57)Abstract:**

PROBLEM TO BE SOLVED: To attain high speed task switching.  
SOLUTION: A sequencer 55 controls the transfer of data between a context saving memory 3 and a data memory and a register file 1 according to the contents of a present task number register 60, next task number register 57, and previous task number register 61 by a start signal from a CPU, and saves/ restores a context. A new task to be switched at the time of requesting task switching is decided by a task permission flag 56 and the present task number register. The register file is constituted of two pairs of register sets in the same constitution, and switched by a register set A using flag 63. A register writing control part writes a context in the register set to be used by the next task, and a register reading control part 68 reads the register set to be used by the present task.

---

Fig. /

【図1】

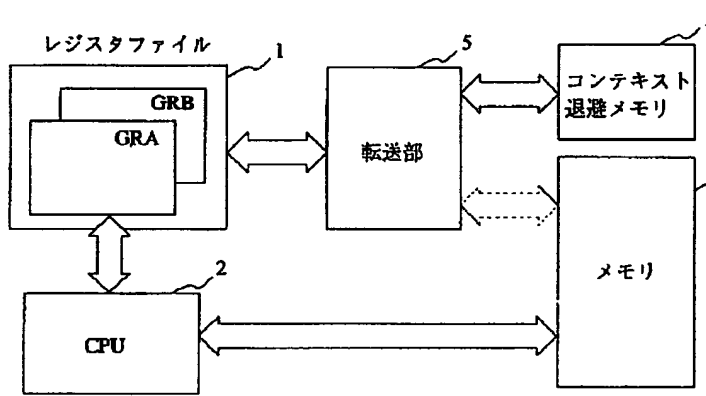


Fig. 10

【図10】

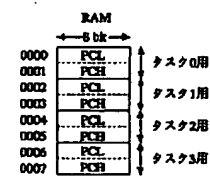
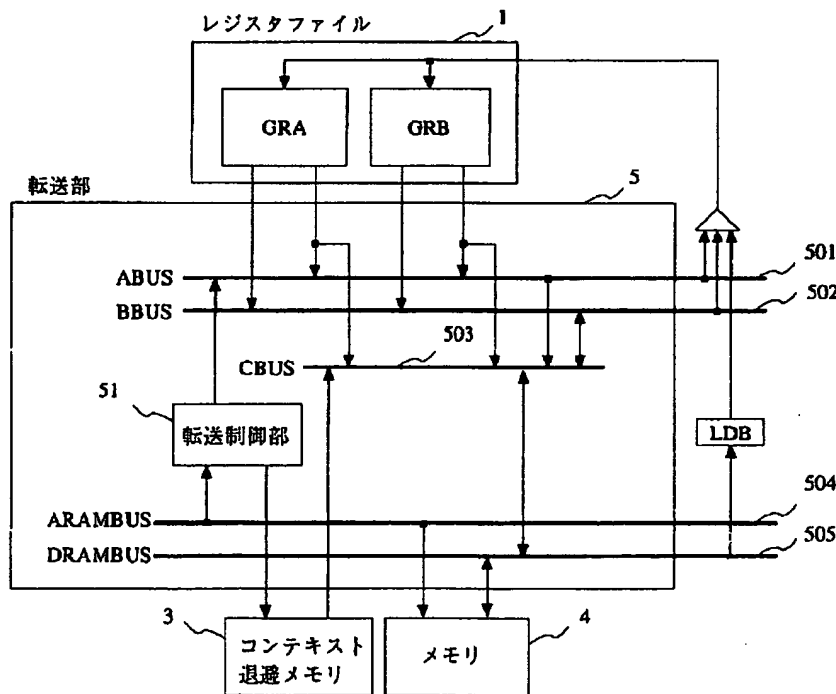


Fig. 2

【図2】



【図3】

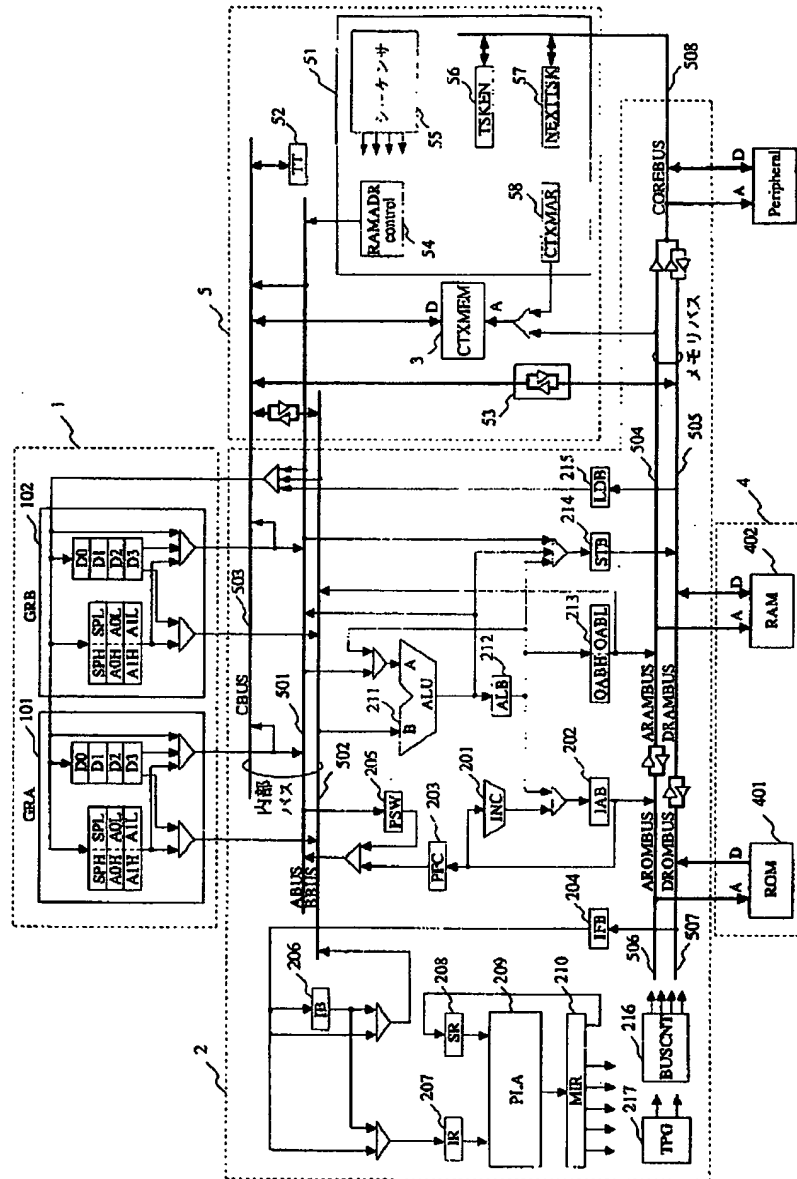


Fig. 3

【図4】

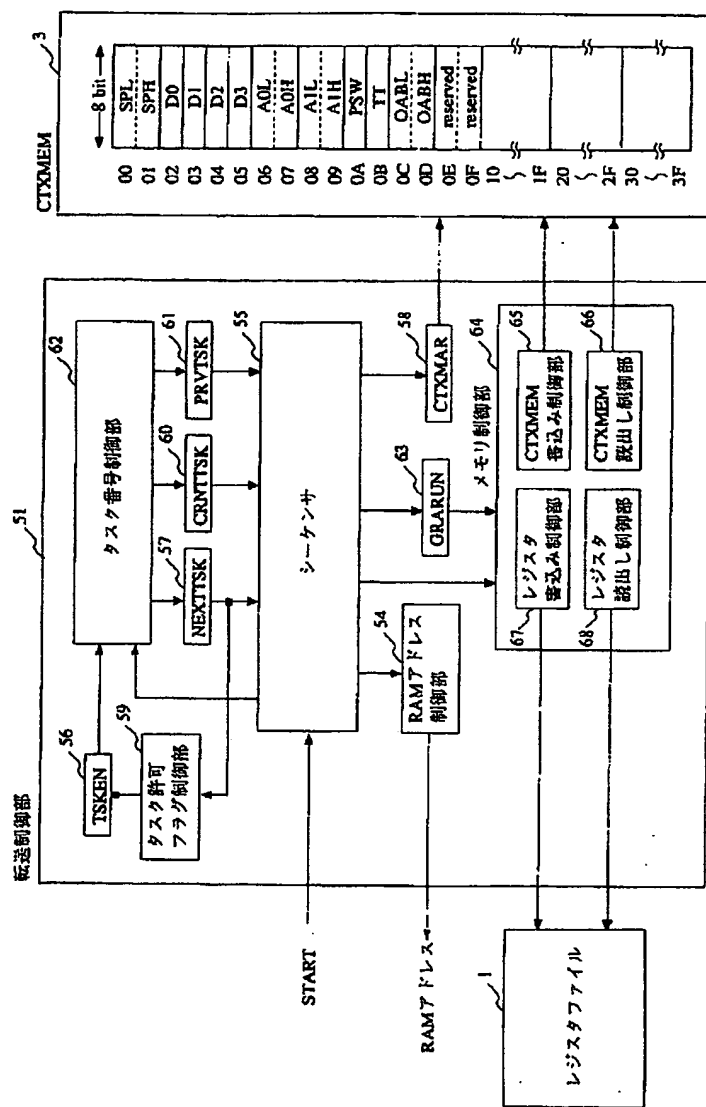
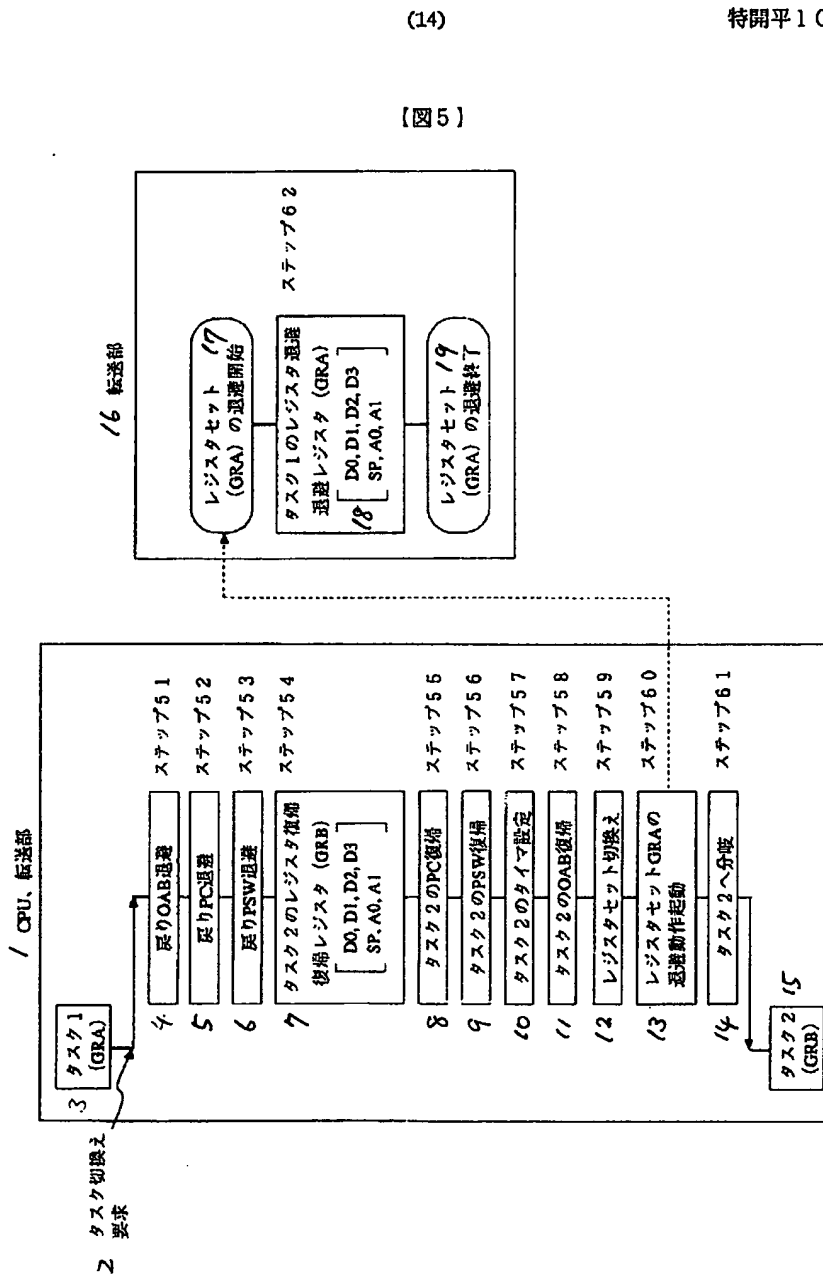


Fig. 4

Fig. 5



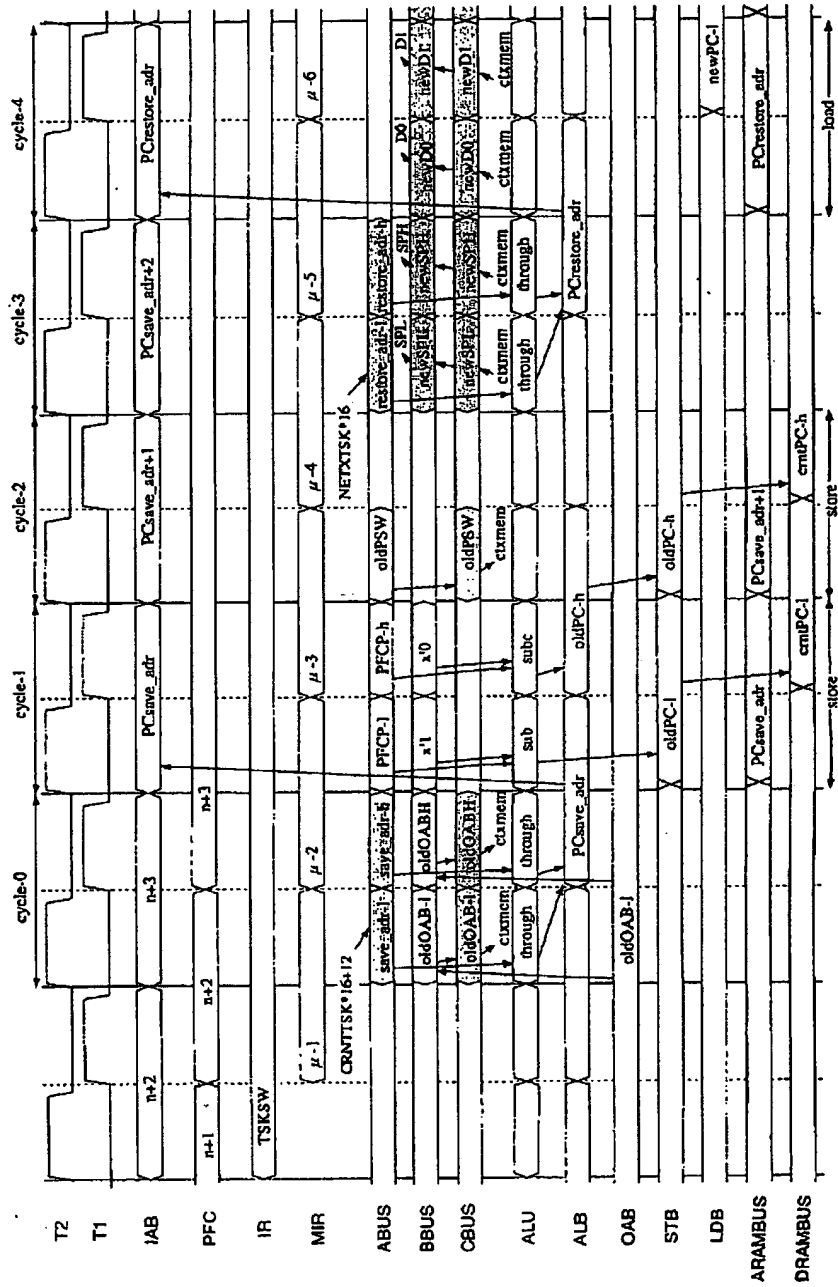
(14)

特開平10-207717

【図5】



【図6】



【図7】

Fig. 7

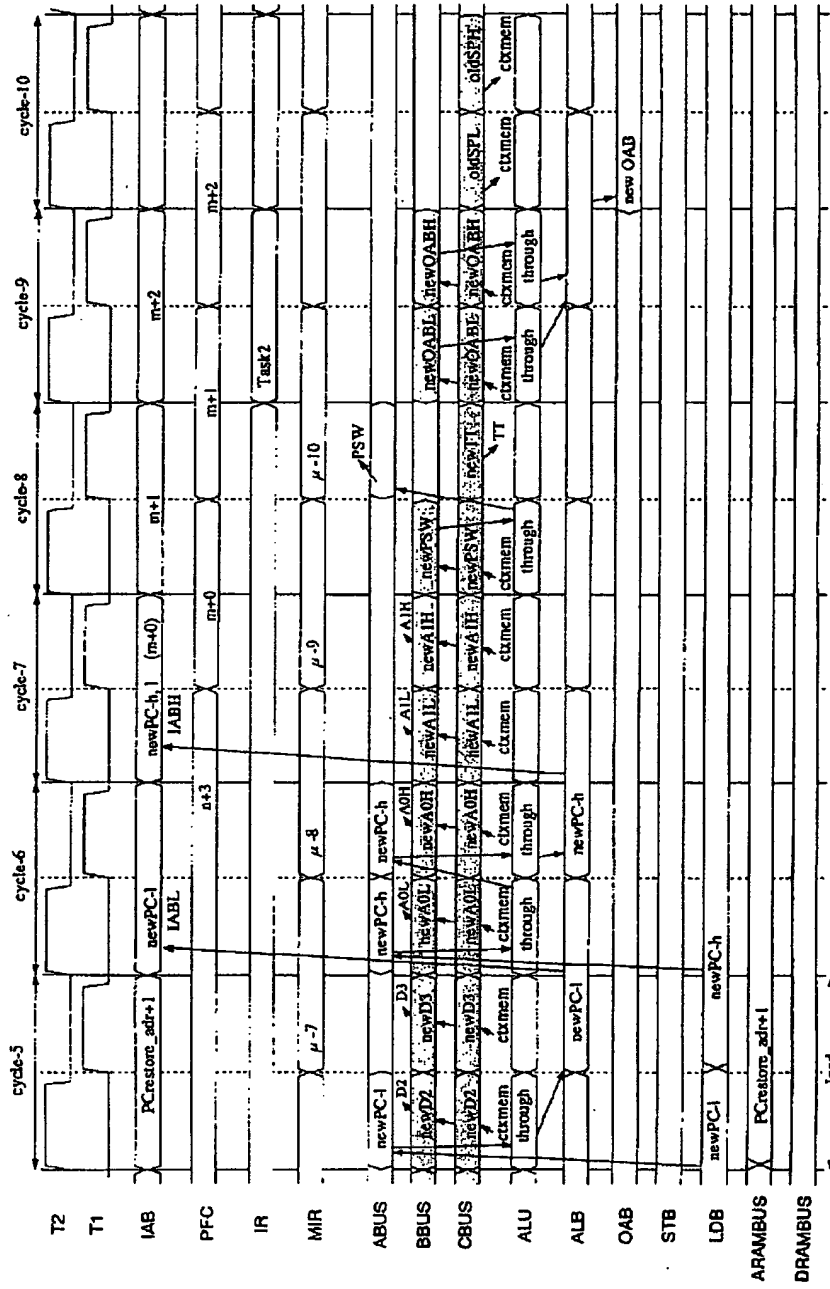
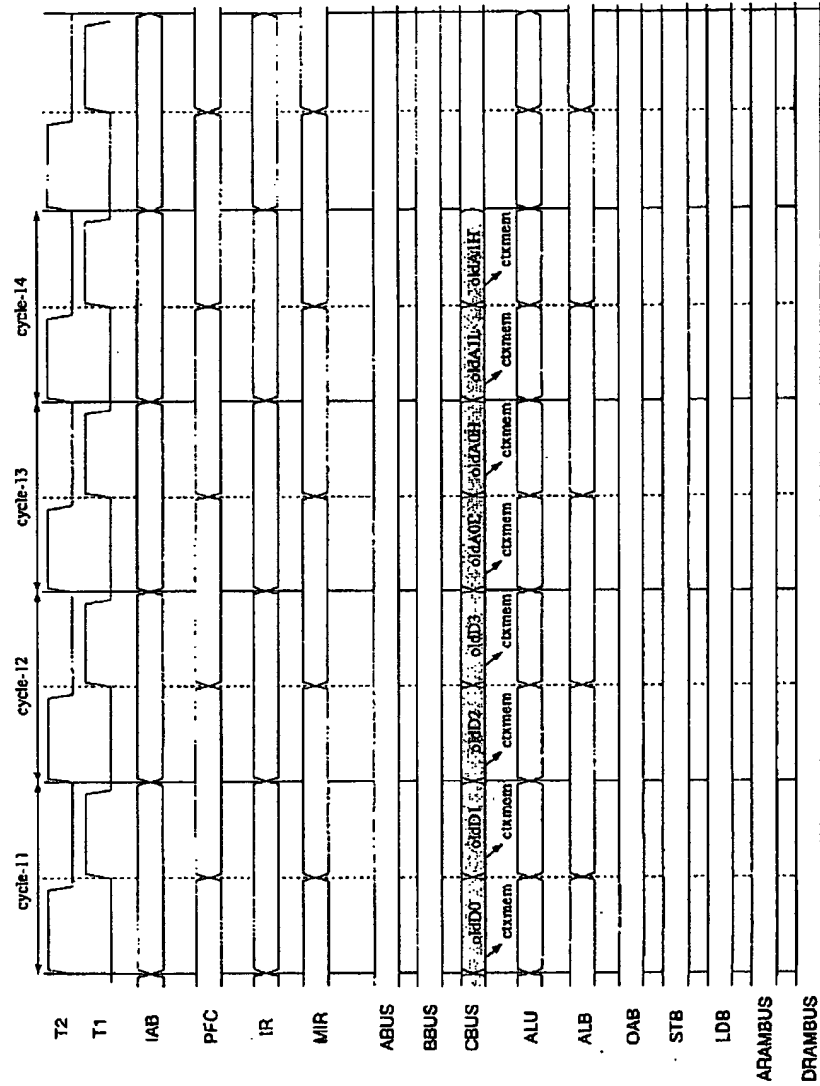


Fig. 8



(17)

特開平10-207717

【図8】

Fig. 9

【図9】

state	timing	CTXMEM			BUS			PRVTSK	CRNTTSK	NEXTTSK	GRARUN
		address	data	R/W	ABUS	BBUS	CBUS				
0	T2	crnttsk+12	oldOABL	W	save_adr-l	oldOABL	(oldOABL)	0	1	2	1
	T1	crnttsk+13	oldOABH	W	save_adr-h	oldOABH	(oldOABH)	1	1	2	1
1	T2							1	1	2	1
	T1							1	1	2	1
2	T2	crnttsk+10	oldPSW	W	oldPSW			1	1	2	1
	T1	nexttsk+0	newSPL	R				1	1	2	1
3	T2	nexttsk+1	newSPH	R	restore_adr-l	(newSPL)	newSPL	1	1	2	1
	T1	nexttsk+2	newD0	R	restore_adr-h	(newSPH)	newSPH	1	1	2	1
4	T2	nexttsk+3	newD1	R		(newD0)	newD0	1	1	2	1
	T1	nexttsk+4	newD2	R		(newD1)	newD1	1	1	2	1
5	T2	nexttsk+5	newD3	R		(newD2)	newD2	1	1	2	1
	T1	nexttsk+6	newAOL	R		(newD3)	newD3	1	1	2	1
6	T2	nexttsk+7	newAOH	R		(newAOL)	newAOL	1	1	2	1
	T1	nexttsk+8	newAIL	R		(newAOH)	newAOH	1	1	2	1
7	T2	nexttsk+9	newAIH	R		(newAIL)	newAIL	1	1	2	1
	T1	nexttsk+10	newPSW	R		(newAIH)	newAIH	1	1	2	1
8	T2	nexttsk+11	newTT	R		(newPSW)	newPSW	1	1	2	1
	T1	nexttsk+12	newOABL	R			newTT	1	1	2	1
9	T2	nexttsk+13	newOABH	R		(newOABL)	newOABL	1	1	2	0
	T1					(newOABH)	newOABH	1	2	3	0
10	T2	prvtsk+0	oldSPL	W			oldSPL	1	2	3	0
	T1	prvtsk+1	oldSPH	W			oldSPH	1	2	3	0
11	T2	prvtsk+2	oldD0	W			oldD0	1	2	3	0
	T1	prvtsk+3	oldD1	W			oldD1	1	2	3	0
12	T2	prvtsk+4	oldD2	W			oldD2	1	2	3	0
	T1	prvtsk+5	oldD3	W			oldD3	1	2	3	0
13	T2	prvtsk+6	oldAOL	W			oldAOL	1	2	3	0
	T1	prvtsk+7	oldAOH	W			oldAOH	1	2	3	0
14	T2	prvtsk+8	oldAIL	W			oldAIL	1	2	3	0
	T1	prvtsk+9	oldAIH	W			oldAIH	1	2	3	0

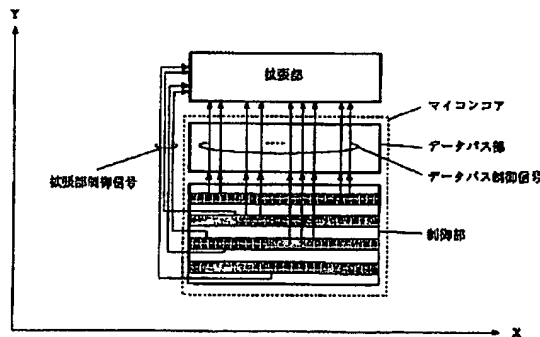
Fig. 11

【図11】

TSKEN	CRNTSK	NEXTSK
0000	0	0
0010	1	1
0011	0	1
	1	0
	2	0
	3	0
0100	0	2
0101	0	2
	1	2
	2	0
	3	0
0110	0	2
	1	2
	2	1
	3	1
0111	0	1
	1	2
	2	0
	3	0
1000	0	3
1001	0	3
	1	3
	2	3
	3	0
1010	0	1
	1	3
	2	3
	3	1

Fig. 13

【図13】



【図12】

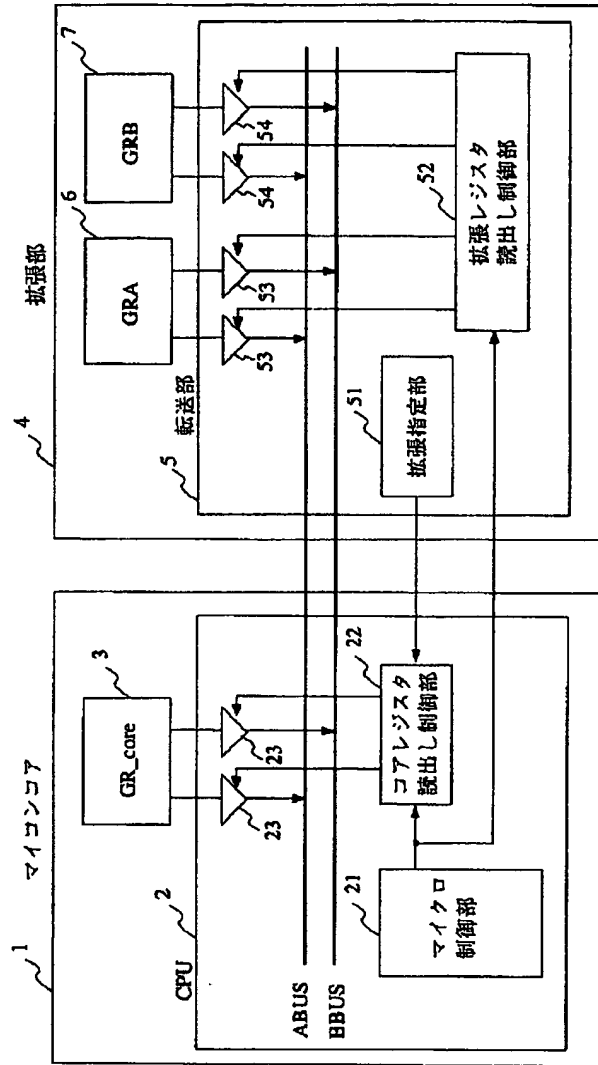


Fig. 12

Fig. 14

【図14】

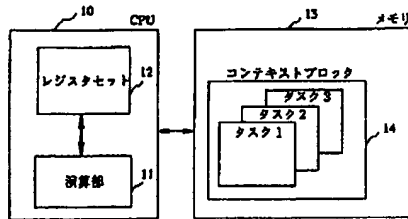
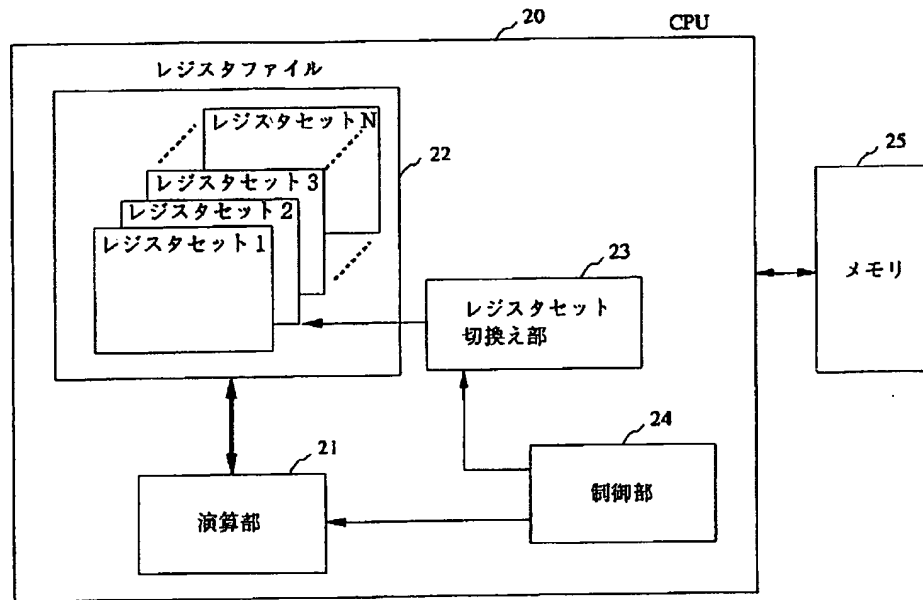


Fig. 15

【図15】





## 【特許請求の範囲】

【請求項1】複数のタスクを順に切り換えながら実行するマイクロコンピュータであって、同一のレジスタ構成で、タスクの実行に1つが使用される少なくとも2つ以上のレジスタセットと、実行中でないタスクのコンテキストデータを保持する第1のメモリと、

CPUがデータ処理に使用する第2のメモリと、前記第1のメモリおよび前記第2のメモリから次に実行するタスクのコンテキストデータをレジスタセットに復帰した時点で、レジスタセットを切り換えるレジスタ切り換え手段と、

前記レジスタセットと前記第1のメモリおよび前記第2のメモリとに接続され、タスク切り換え時にタスクのコンテキストデータの一部を前記第2のメモリに転送し、切り換えられる前に使用していたレジスタセットの内容を新規タスクの実行中に前記第1のメモリに退避する転送手段とを備えることを特徴とするマイクロコンピュータ。

【請求項2】前記転送手段は、CPUからの起動要求により動作を開始するシーケンサと、タスクが使用していたレジスタセットのデータを読み出す読み出し手段と、

レジスタ読み出し手段により読み出したデータを前記第1のメモリに書き込む書き込み手段と、

次に実行するタスクのコンテキストデータを前記第1のメモリから読み出すメモリ読み出し手段と、

メモリ読み出し手段により読み出したデータを次に実行するタスクが使用するレジスタセットに書き込むレジスタ書き込み手段と、

前記シーケンサに接続され、前記メモリ書き込み手段または前記メモリ読み出し手段がアクセスするアドレスを指定する第1のアドレスレジスタと、

前記シーケンサに接続され、前記第2のメモリのアドレスを指定する第2のアドレスレジスタと、

前記シーケンサに接続され、タスクが使用中のレジスタセットを指定するレジスタセット番号指定手段とを備えることを特徴とする請求項1記載のマイクロコンピュータ。

【請求項3】前記転送手段は、

現在実行中のタスクの番号を指定する現タスク番号レジスタと、

次に実行するタスクの番号を指定する次タスク番号レジスタと、

前回実行したタスクの番号を指定する前タスク番号レジスタと、

前記シーケンサのステートに基づいて、前記現タスク番号レジスタの内容を前記前タスク番号レジスタに転送し、前記次タスク番号レジスタの内容を前記現タスク番号レジスタに転送するタスク番号制御手段とを備えるこ

とを特徴とする請求項2記載のマイクロコンピュータ。

【請求項4】前記転送手段は、

前記レジスタセットと前記第1のメモリとに接続され、かつデータバスを介して前記第2のメモリとに接続される内部バスを備え、CPUが前記第2のメモリを使用中にレジスタセットの読み出しと前記第1のメモリへの書き込みを並行処理することを特徴とする請求項1記載のマイクロコンピュータ。

【請求項5】前記転送手段は、コンテキストの一部を前記第2のメモリのあらかじめ割り当てられた特定番地に格納することを特徴とする請求項1記載のマイクロコンピュータ。

【請求項6】前記転送手段は、コンテキストの一部を前記第2のメモリに割り当てられたスタック領域に格納することを特徴とする請求項1記載のマイクロコンピュータ。

【請求項7】前記レジスタセットは、CPUが使用するスタックの最終位置を示すスタックポインタを備え、前記スタックポインタは、コンテキストを前記第2のメモリのスタック領域に格納後、スタックの最終位置とは異なるアドレスを保持することを特徴とする請求項6記載のマイクロコンピュータ。

【請求項8】複数のタスクを順に切り換えながら実行するマイクロコンピュータであって、

同一のレジスタ構成で、タスクの実行に1つが使用される少なくとも2つ以上のレジスタセットと、

実行中でないタスクのコンテキストデータを保持する第1のメモリと、

CPUがデータ処理に使用する第2のメモリと、

前記第1のメモリおよび前記第2のメモリから次に実行するタスクのコンテキストデータをレジスタセットに復帰した時点で、レジスタセットを切り換えるレジスタ切り換え手段と、

前記レジスタセットと前記第1のメモリおよび前記第2のメモリとに接続され、タスク切り換え時にタスクのコンテキストデータの一部を前記第2のメモリに転送し、切り換えられる前に使用していたレジスタセットの内容を新規タスクの実行中に前記第1のメモリに退避する転送手段と、

命令を処理するマイクロ制御手段と、

前記マイクロ制御手段が出力する制御信号と内部のバスとを出力するCPUと、

CPU内部にあって、前記マイクロ制御手段によりレジスタセットを内部バスに読み出すレジスタ読み出し手段と、

前記CPUの制御信号と内部バスとに接続され、前記CPUと同一のレジスタセットを備える拡張手段と、

前記拡張手段が前記CPUに接続されていることを通知し、前記読み出し手段の動作を停止させる拡張指定手段とを備えたことを特徴とするマイクロコンピュータ。



【請求項9】前記CPUはデータを処理するデータベース部と前記データベースを制御する制御部とからなり、前記制御部は前記データベースを制御するデータベース制御信号と、前記拡張部を制御する拡張部制御信号を出力し、データベース制御信号はデータベース部のデータ処理の流れとは垂直方向に配置されてデータベース部および前記拡張部に

入力され、  
拡張部制御信号は、データベース部のデータ処理の流れと水平方向に配置されて制御部から出力された後に、前記拡張部に入力されることを特徴とする請求項8記載のマイクロコンピュータ。

【請求項10】複数のタスクを順に切り換えながら実行するマイクロコンピュータであって、  
n個のタスクの起動を許可するか禁止するかを示すnビットの第1のフラグを備え、前記第1のフラグが許可されたタスクのみをタスク番号の昇順または降順に順次実行することを特徴とするマイクロコンピュータ。

【請求項11】次に実行する予定のタスク番号を示すmビットであってかつ、mをべきとする2のべき乗値が前記第1のフラグの値以下である第2のフラグと、  
前記第2のフラグに所望のタスク番号を設定することで、前記第1のフラグの該当するビットをセットするタスク許可フラグ制御手段とを備え、  
所望のタスクを直接起動することを特徴とする請求項10記載のマイクロコンピュータ。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、複数のタスクを切り換えながら実行するマイクロコンピュータに関する。

【0002】

【従来の技術】近年、マイクロコンピュータはあらゆる機器に利用されており、処理能力の向上に伴い、より広い分野でより複雑な制御に応用されている。組み込み用途ではより複雑な制御に応用できるように、マルチタスク処理などの高度な機能が望まれている。

【0003】図14は、第1の従来例におけるマルチタスク機能を有するマイクロコンピュータの説明図である。同図において、CPU10は、演算部11とレジスタセット12からなる。

【0004】演算部11は、タスクを1つずつ実行し、レジスタセット12は、タスク実行に必要なデータを保持する。

【0005】メモリ13は、各タスク毎にコンテキスト（レジスタセットのデータ、PCの内容、PSWの内容等のタスクの実行によって変化するタスク固有のデータ）を退避する領域であるコンテキストブロック14を有する。同図では、コンテキストブロック14中にタスク1～3用のコンテキスト退避領域が示されている。

【0006】第1の従来例におけるタスクはソフトウェア（OS）によって管理される。例えば、タスク1から

タスク2に切り換わる場合、OSは、タスク切り換えを示すイベント（タイマ割込み等）を受け付けると、まずCPU10からメモリ13にタスク1のコンテキストを退避し、次にメモリ13からCPU10にタスク2のコンテキストを復帰してからタスク2の実行を開始する。

【0007】上記のように、第1の従来例では、ソフトウェア（OS）によりマルチタスクを実現しているので、ハードウェア規模が小さくて済むという利点がある。

【0008】図15は、第2の従来例におけるマルチタスク機能を有するマイクロコンピュータの説明図である。同図においてCPU20は、演算部21、レジスタファイル22、レジスタセット切り換え部23、制御部24とからなる。

【0009】演算部21は、タスクを1つずつ実行し、レジスタファイル22は、タスクの実行に必要なデータを保持する複数のレジスタセットからなる。

【0010】レジスタセット切り換え部23は、現在実行中のタスクに対応するレジスタセットを有効にする。この有効なレジスタセットのみが演算部21に使用される。制御部24は、タスクの切り換えを制御する。

【0011】メモリ25は、レジスタセットのデータを除くコンテキストの退避領域を有する。この第2の従来例におけるタスクはハードウェアによって管理される。例えば、タスク1からタスク2に切り換える場合、制御部24は、タスク切り換えを指示するイベント（タイマ割込み等）を受け付けると、レジスタセット切り換え部23にタスク2に対応するレジスタセットを有効にするように指示すると共に、演算部21のPCの内容、PSWの内容等をメモリ25に退避し、タスク2のPCの内容、PSWの内容等を演算部21に復帰する。その後、演算部21は、有効なレジスタセット2を用いてタスク2を実行する。

【0012】上記のように第2の従来例では、ハードウェアによってマルチタスクを実現しているので、タスクの切り換えが高速であるという利点がある。

【0013】

【発明が解決しようとする課題】しかしながら、上記従来例によれば、組み込み用途のマイクロコンピュータにおけるマルチタスク管理には適していないという問題があった。

【0014】第1の従来例によれば、タスク切り換えに際してコンテキストの退避および復帰をソフトウェアによって実現するので切り換え速度が遅く、リアルタイム性を要求される制御の組み込み用途のマイクロコンピュータには適していない。また、第2の従来例によれば、タスク数と同数のレジスタセットを必要とし、しかもタスク切り換えをハードウェアで実現するので制御部のハードウェア規模が大きくなる。その結果コストが高くなり組み込み用途のマイクロコンピュータには適していな

10

20

30

40

50

い。

【0015】本発明は上記の問題点に鑑み、コストを増大させることなく高速にタスクを切り換えるマイクロコンピュータを提供することを目的としている。

【0016】

【課題を解決するための手段】上記課題を解決するため、請求項1の発明は、複数のタスクを順に切り換えながら実行するマイクロコンピュータであって、同一のレジスタ構成で、タスクの実行に1つが使用される少なくとも2つ以上のレジスタセットと、実行中でないタスクのコンテキストデータを保持する第1のメモリと、CPUがデータ処理に使用する第2のメモリと、前記第1のメモリおよび前記第2のメモリから次に実行するタスクのコンテキストデータをレジスタセットに復帰した時点で、レジスタセットを切り換えるレジスタ切り換え手段と、前記レジスタセットと前記第1のメモリおよび前記第2のメモリとに接続され、タスク切り換え時にタスクのコンテキストデータの一部を前記第2のメモリに転送し、切り換えられる前に使用していたレジスタセットの内容を新規タスクの実行中に前記第1のメモリに退避する転送手段とを有する。

【0017】請求項2の発明は、請求項1において前記転送手段が、CPUからの起動要求により動作を開始するシーケンサと、タスクが使用していたレジスタセットのデータを読み出す読み出し手段と、レジスタ読み出し手段により読み出したデータを前記第1のメモリに書き込む書き込み手段と、次に実行するタスクのコンテキストデータを前記第1のメモリから読み出すメモリ読み出し手段と、メモリ読み出し手段により読み出したデータを次に実行するタスクが使用するレジスタセットに書き込むレジスタ書き込み手段と、前記シーケンサに接続され、前記メモリ書き込み手段または前記メモリ読み出し手段がアクセスするアドレスを指定する第1のアドレスレジスタと、前記シーケンサに接続され、前記第2のメモリのアドレスを指定する第2のアドレスレジスタと、前記シーケンサに接続され、タスクが使用中のレジスタセットを指定するレジスタセット番号指定手段とを有する。

【0018】請求項3の発明は、請求項2において前記転送手段が、現在実行中のタスクの番号を指定する現タスク番号レジスタと、次に実行するタスクの番号を指定する次タスク番号レジスタと、前回実行したタスクの番号を指定する前タスク番号レジスタと、前記シーケンサのステートに基づいて、前記現タスク番号レジスタの内容を前記前タスク番号レジスタに転送し、前記次タスク番号レジスタの内容を前記現タスク番号レジスタに転送するタスク番号制御手段とを有する。

【0019】請求項4の発明は、請求項1において前記転送手段が、前記レジスタセットと前記第1のメモリとに接続され、かつデータバスを介して前記第2のメモリ

とに接続される内部バスを有する。

【0020】請求項5の発明は、請求項1において前記転送手段が、コンテキストの一部を格納する前記第2のメモリのあらかじめ割り当てられた特定番地を有する。

【0021】請求項6の発明は、請求項1において前記転送手段が、コンテキストの一部を格納する前記第2のメモリに割り当てられたスタック領域を有する。

【0022】請求項7の発明は、請求項6において前記レジスタセットが、コンテキストを前記第2のメモリのスタック領域に格納後、スタックの最終位置とは異なるアドレスを保持するスタックポインタを有する。

【0023】請求項8の発明は、複数のタスクを順に切り換えながら実行するマイクロコンピュータであって、同一のレジスタ構成で、タスクの実行に1つが使用される少なくとも2つ以上のレジスタセットと、実行中でないタスクのコンテキストデータを保持する第1のメモリと、CPUがデータ処理に使用する第2のメモリと、前記第1のメモリおよび前記第2のメモリから次に実行するタスクのコンテキストデータをレジスタセットに復帰した時点で、レジスタセットを切り換えるレジスタ切り換え手段と、前記レジスタセットと前記第1のメモリおよび前記第2のメモリとに接続され、タスク切り換え時にタスクのコンテキストデータの一部を前記第2のメモリに転送し、切り換えられる前に使用していたレジスタセットの内容を新規タスクの実行中に前記第1のメモリに退避する転送手段と、命令を処理するマイクロ制御手段と、前記マイクロ制御手段が出力する制御信号と内部のバスとを出力するCPUと、CPU内部にあって、前記マイクロ制御手段によりレジスタセットを内部バスに読み出すレジスタ読み出し手段と、前記CPUの制御信号と内部バスとに接続され、前記CPUと同一のレジスタセットを備える拡張手段と、前記拡張手段が前記CPUに接続されていることを通知し、前記読み出し手段の動作を停止させる拡張指定手段とを有する。

【0024】請求項9の発明は、請求項8においてCPUはデータを処理するデータバス部とデータバスを制御する制御部とからなり、前記制御部はデータバス部のデータ処理の流れとは垂直方向に配置されてデータバス部および前記拡張部に入力される前記データバスを制御するデータバス制御信号と、データバス部のデータ処理の流れと水平方向に配置されて制御部から出力された後に、前記拡張部に入力される前記拡張部を制御する拡張部制御信号を有する。

【0025】請求項10の発明は、複数のタスクを順に切り換えながら実行するマイクロコンピュータであって、n個のタスクの起動を許可するか禁止するかを示すnビットの第1のフラグを有する。

【0026】請求項11の発明は、請求項10において、次に実行する予定のタスク番号を示すmビットであってかつ、mをべきとする2のべき乗値が前記第1のフ

ラグの値以下である第2のフラグと、前記第2のフラグに所望のタスク番号を設定することで、前記第1のフラグの該当するビットをセットするタスク許可フラグ制御手段を有する。

【0027】

【発明の実施の形態】図1は、本発明の実施例におけるマイクロコンピュータの主要な構成を示すブロック図である。実施例のマイクロコンピュータは、レジスタファイル1、CPU2、コンテキスト退避メモリ3、メモリ4、転送部5を有する。

【0028】レジスタファイル1は、レジスタセットA（GRA）とレジスタセットB（GAB）とからなる。レジスタセットA、Bはタスク切り換え毎に交互に切り換えられ、常にどちらか一方がCPU2のタスク実行に使用される。

【0029】CPU2は、複数のタスクを切り換えながら実行する。CPU2が実行するタスクは、タスク切り換えを指示するイベントが発生する毎に、転送部5のタスク許可フラグに基づいてタスク番号の昇順又は降順に切り換えられる。タスク切り換えのイベントは例えば、

【0030】コンテキスト退避メモリ3は、CPU2が実行中でないタスクの情報を退避しておくメモリである。タスクの情報はレジスタセットの内容、プロセッサステータスワード（PSW）の内容、プログラムカウンタ（PC）の内容等を含むコンテキストである。

【0031】メモリ4は、CPU2がタスクの実行で処理するデータ等を記憶する。転送部5は、タスク切り換えイベントが発生する毎に、CPU2が次に実行するタスクのコンテキストをコンテキスト退避メモリ3から使用中でないレジスタセットに復帰し、レジスタセットを切り換える。切り換えたレジスタセットを使用してCPU2が次のタスクを実行中に、切り換える前に使用中であったレジスタセットの内容をコンテキスト退避メモリ3に退避する。

【0032】図2は、図1に示したマイクロコンピュータの転送部の内部バスを含むより詳細な構成を示すブロック図である。転送部5は、転送制御部51、ABUS501、BBUS502、CBUS503、ARAMBUS504、DRAMBUS505から構成される。

【0033】転送制御部51は、転送部5の各バスや各バスに接続されたメモリ、レジスタファイル等を制御する。転送制御部51は、CPUがコンテキスト退避メモリ3をアクセスする際のアドレスをARAMBUS504から入力する。さらに、ABUS501にコンテキスト退避時の特定アドレスを出力する。

【0034】図3は、図1に示したマイクロコンピュータのより詳細な構成を示すブロック図である。

【0035】同図に示すように、本マイクロコンピュ

タは、レジスタセットA101、レジスタセットB102、インクリメンタ201、命令アドレスバッファ202、プリフェッチカウンタ203、命令フェッチバッファ204、プロセッサステータスワード205、命令バッファ206、命令レジスタ207、ステータスレジスタ208、PLA209、マイクロ命令レジスタ210、ALU211、ALU出力バッファ212、オペランドアドレスバッファ213、ストアバッファ214、ロードバッファ215、バス制御部216、タイミング生成部217、ROM401、RAM402、転送制御部51、タスクタイマ52、バススイッチ53、RAMアドレス生成部54、シーケンサ55、タスク許可フラグ56、次タスク番号レジスタ57、コンテキスト退避メモリアドレスレジスタ58、ABUS501、BBUS502、CBUS503、ARAMBUS504、DRAMBUS505、AROMBUS506、DROMBUS507、COREBUS508とからなる。

【0036】レジスタセット101とレジスタセット102とは、同一のレジスタ構成であり、それぞれ8ビット幅のデータレジスタ（D0、D1、D2、D3）と16ビット幅のアドレスレジスタ（A0、A1）と、16ビット幅のスタックポインタ（SP）と、これらのデータを選択して出力するセレクタとを有する。インクリメンタ201、命令アドレスバッファ202、プリフェッチカウンタ203、命令フェッチバッファ204からなる回路は、ROM401から命令を順次プリフェッチする命令先読み準備部である。

【0037】プロセッサステータスワード205は、ALU211の演算フラグやCPU2の割込み状態等を記憶する。

【0038】命令バッファ206は、命令先読み準備部によってプリフェッチされた複数の命令を保持する命令キューである。

【0039】命令レジスタ207、ステータスレジスタ208、PLA209、マイクロ命令レジスタ210からなる回路は、命令バッファ206から供給される命令を解釈し、実行を制御する命令制御部である。また、この命令制御部はタスク切り換えを指示するイベントが発生すると、タスクのコンテキストの一部を退避、復帰する制御および転送制御部51に転送の開始を通知する制御を行う。本実施例では、上記イベントはタスクタイマ52による割込みとする。また、命令制御部は、ABUS501、BBUS502のデータの入出力を制御する。

【0040】ALU211、ALU出力バッファ212、オペランドアドレスバッファ213、ストアバッファ214、ロードバッファ215からなる回路は、命令制御部の制御に従って命令を実行する命令実行部である。

【0041】ROM401とRAM402は図1に示し

10

20

30

40

50

たメモリ4に相当し、複数のタスクのプログラム、スタック領域、タスク開始アドレス退避領域を有する。各タスクのプログラムはROM401に記憶され、スタック領域、タスク開始アドレス退避領域はRAM402に設けられる。

【0042】CPU内部バスはABUS501、BBUS502、CBUS503の3本から構成され、CPU外部のメモリバスは、ARAMBUS504、DRAMBUS505、AROMBUS506、DROMBUS507、COREBUS508の5本から構成される。

【0043】ABUS501、BBUS502は、主にレジスタファイル1に対してアクセスされるデータを伝達する。

【0044】CBUS503は、主にコンテキスト退避メモリ3に対してアクセスするデータを伝達する。

【0045】ARAMBUS504とDRAMBUS505は、それぞれ主にRAM402に対してアクセスするアドレスとデータを伝達し、AROMBUS506とDROMBUS507は、それぞれ主にROM401に対してアクセスするアドレスとデータを伝達する。

【0046】COREBUS508は、主にマイクロコンピュータの周辺機能に含まれるレジスタ等をアクセスするアドレスとデータを伝達する。

【0047】タスクタイマー52は、CPUが実行中のタスクの時間を計測するタイマーで、タスク実行開始時に設定され、一定時間間隔でタイマーの値がデクリメントされる。タスクタイマーの内容がゼロになると、CPU2に対してタスク切り換えイベントの割込みを発生する。タスクタイマーに設定される値は、あらかじめコンテキスト退避メモリに設定されている。また、タスク切り換え時にはタスクタイマーの内容は常にゼロであるため、コンテキスト退避メモリには退避されない。

【0048】バススイッチ53は、DRAMBUS505とCBUS503との間でデータの受け渡しをするバッファである。

【0049】RAMアドレス生成部54は、タスク切り換え時にCPUが退避、復帰するコンテキストの一部を格納するRAM402のアドレスを生成する。

【0050】シーケンサ55は、CPUからの起動要求に応じて、CBUS503の制御や転送制御部内のレジスタの更新を制御する信号を出力し、タスクのコンテキストを退避、復帰させる。

【0051】タスク許可フラグ56は、CPUが実行するタスクの数と同数のビット数から構成され、各ビットに対応するタスクの実行を許可するか、またはタスクの実行を禁止するかを指定する。

【0052】次タスク番号レジスタ57は、タスク切り換えのイベントが発生したときに、次に切り換えて実行するタスクの番号を格納するレジスタで、レジスタの内容は、シーケンサがタスク許可フラグ56の内容に基づ

いて決められる。

【0053】タスク許可レジスタ56および次タスク番号レジスタ57は、COREBUS508にも接続されており、CPUのプログラム実行によって内容の読み出しまたは書き込みが出来る。

【0054】コンテキスト退避メモリアドレスレジスタ58は、コンテキスト退避メモリ3をアクセスする番地を格納するレジスタで、シーケンサ55がコンテキスト退避メモリをアクセスする際に参照される。

【0055】図4は、図2のマイクロコンピュータの転送制御部51のより詳細な構成と、コンテキスト退避メモリ3のより詳細なデータ構成を示すブロック図である。同図において、タスク許可フラグ制御部59は、次タスク番号レジスタ57に格納されたタスクの番号をデコードし、タスク許可フラグ56のタスク番号に対応するビットを1にする。また、命令によりタスク許可フラグを変更する場合、タスク許可フラグ制御部59はCOREBUS508のデータをタスク許可フラグ56に設定する。

【0056】現タスク番号レジスタ60は、CPUが現在実行中のタスクの番号を格納する。前タスク番号レジスタ61は、CPUが現在実行中のタスクに切り換わる前に実行していたタスクの番号を格納する。

【0057】タスク番号制御部62は、タスク許可フラグ56の内容と現タスク番号レジスタ60の内容とシーケンサ55が出力するレジスタ書換信号とを入力し、次タスク番号レジスタ57、現タスク番号レジスタ60、前タスク番号レジスタ61の内容をそれぞれ更新する。シーケンサ55が出力する書換信号のタイミングは、図9に矢印で示されている。また、タスク番号制御部が次タスク番号レジスタを更新する内容は、図11に示されている。

【0058】図9によると、矢印で示されているように、前タスク番号レジスタ61が書き換えられるのはシーケンサ55のステート0であり、次タスク番号レジスタ57および現タスク番号レジスタ60が書き換えられるのはシーケンサ55のステート9である。

【0059】レジスタセットA使用フラグ63は、CPUがタスク実行に使用するレジスタセットを指定するフラグであり、レジスタセットA使用フラグが1の時、レジスタセットAを使用し、0の時、レジスタBを使用する。

【0060】メモリ制御部64は、コンテキスト退避メモリ3の読み出しおよび書き込みと、レジスタファイル1の読み出しおよび書き込みを制御する。メモリ制御部64は、コンテキスト退避メモリ3へのデータの書き込みを制御するコンテキスト退避メモリ書き込み制御部65と、コンテキスト退避メモリ3からのデータの読み出しを制御するコンテキスト退避メモリ読み出し制御部68と、レジスタファイルへのデータの書き込みを制御す

るレジスタ書き込み制御部67と、レジスタファイルからのデータの読み出しを制御するレジスタ読み出し制御部68とを有する。

【0061】図4に示すコンテキスト退避メモリ3は、タスク数4までに対応する場合の構成を示し、64バイトの容量である。64バイトを4つに分割し、分割した各領域のアドレスの若いほうから順に、スタックポインタ(SP)、データレジスタ(D0、D1、D2、D3)、アドレスレジスタ(A0、A1)、プロセッサステータスワード(PSW)、タスクタイマー(TT)、オペランドアドレスバッファ(OAB)を退避する。16ビット幅のレジスタは、下位の8ビットがアドレスの若いほうに退避される。また、本実施例では分割した各領域には、2バイトの予備の退避領域が存在する。

【0062】図5は、タスク切り換え時の処理のフローチャートである。同図において、「CPU、転送部」、「転送部」はそれぞれ、タスク切り換え時にCPU2および転送部5が同期して並行に処理するフローと転送部5単独で処理するフローを示す。

【0063】同図のフローチャートは、CPU2がレジスタセットAを使用してタスク1を実行中に、タスク切り換え要求が発生し、タスク2に切り換わるフローを示している。CPU2は、タスク切り換え要求が発生すると、実行中のタスクを一次停止し、タスク切り換え処理に移行する。このタスク切り換えは、CPU2は割込み処理と同様にPLA209により処理され、転送部5はシーケンサ55により処理される。

【0064】図6、図7、図8は、タスク切り換え処理のタイミングチャートを示す図である。同図には、タイミング(T2、T1)、IAB202、PFC203、IR207、MIR210、ABUS501、BBUS502、CBUS503、ALU211、ALB212、OAB213、STB214、LDB215、ARAMBUS504、DRAMBUS505の内容が、各サイクル毎にそれぞれ示されている。なお同図において右方向が時間軸である。

【0065】また、ABUS、BBUS、CBUSの網かけは転送部5により制御されていることを示している。

【0066】図9は、転送部5のシーケンサ55による各部の動作を示した図であり、シーケンサ55のステートおよびタイミング毎に、コンテキスト退避メモリ3のアドレス(address)とデータ(data)と転送方向(R/W)、バス(ABUS、BBUS、CBUS)、タスク番号レジスタ(PRVTSK、CRNTTSK、NEXTTSK)、レジスタファイルA使用フラグ(GRARUN)の状態がそれぞれ図示されている。

【0067】転送方向のWはコンテキスト退避メモリへの書き込みを、Rはコンテキスト退避メモリからの読み出しをそれぞれ示している。

【0068】同図のバスの欄にカッコ付きで示されたものは、他のバスに出力された内容が転送されていることを示す。また、「old」はタスク切り換え要求発生時点で実行していたタスクのコンテキストを指し、「new」はタスク切り換え要求により新しく切り換わるタスクのコンテキストを指す。「address」の欄の「crnttsk」、「nexttsk」、「prvtsk」はそれぞれ、現タスク番号レジスタCRNTTSK60、次タスク番号レジスタNEXTTSK57、前タスク番号レジスタPRVTSK61の内容を16倍した値を示し、例えば、ステート0の「crnttsk+12」はCRNTTSKの内容「1」を16倍した値に12を加算した値「28」をコンテキスト退避メモリのアドレスにすることを示している。

【0069】同図のステートおよびタイミングは、図6Aから図6Cのサイクルの表示に一致している。従って、例えば図7のステート0は、図6Aのサイクル0に対応し、タイミングT2、タイミングT1は、図6のT2の状態1、T1の状態1にそれぞれ対応している。

【0070】図10は、各タスクのプログラムカウンタを退避するRAM402の特定領域の構成を示した図である。本実施例では4つのタスクのプログラムカウンタの内容を退避する領域が、アドレス0からアドレス7までの8バイトの特定領域に設けられている。

【0071】以下、図5、図6、図7、図8、図9に基づいてタスク切り換え処理をステートに従って詳細に説明する。

【0072】<ステート0>CPU2はタスク切り換え要求発生時のOAB213の内容をタイミングT2、T1でそれぞれBBUSに出力する。

【0073】転送部5は、現タスク番号レジスタの内容が1なので、ABUSにはPCの退避アドレス2番地の下位8ビットをタイミングT2で、上位8ビットをタイミングT1でそれぞれABUSに出力するとともに、CPUがBBUSに出力したOABの内容をCBUSに転送し、コンテキスト退避メモリの28番地と29番地に書き込む(ステップ51)。また、CRNTTSKの内容をPRVTSKに転送する。

【0074】ABUSに出力された退避アドレスは、CPUによりALUを通してALB212に転送される。

【0075】<ステート1>CPUはALBに格納された退避アドレスをIAB202に転送するとともに、ARAMBUSを通して、ARAMBUSに出力する。また、ABUSにPFC203の内容を、BBUSに定数「1」をそれぞれ出力し、実行中であったタスク1の戻り番地(プログラムカウンタの値)をALUにより計算する。計算結果の下位8ビットはSTB214に転送され、タイミングT2でDRAMBUSに出力される。計算結果の上位8ビットはALBに格納される。従って、

CPUはRAM402の2番地に戻り番地の下位8ビッ

トを書き込む(ステップ52)。

【0076】転送部はステート1では何もしない。なお、戻り番地の計算は、計算時の命令バッファ206の先読み量によって制御されるが、本発明には関係ないので詳細な説明は省略する。

【0077】<ステート2>CPUはサイクル1で使用したIABの内容をインクリメンタ201により+1し、AROMBUSを通してARAMBUSに出力するとともに、サイクル1で計算したタスク1の戻り番地の上位8ビットをDRAMBUSに出力して、RAM402の3番地に書き込む(ステップ52)。また、タスク1のPSWの内容をABUSに出力する。

【0078】転送部はタイミングT2でABUSに出力されたPSWの内容を、CBUSに転送し、コンテキスト退避メモリの26番地に書き込み(ステップ53)、タイミングT1でコンテキスト退避メモリの32番地を読み出す。また、転送部がアクセスするレジスタセットは、レジスタセットA使用フラグの内容を反転した値が示すレジスタセットを使用する。従って、図7の例では、ステート2よりステート8の間では、転送部はレジスタセットBに対してアクセスし、ステート9よりステート14の間では、レジスタセットAに対してアクセスする。

【0079】<ステート3>CPUは、ABUSに出力されたタスク2の戻り番地が退避されているRAM402のアドレスをALUを通してALBに格納する。

【0080】転送部は、次タスク番号レジスタの内容「2」に基づいて、タスク2の戻り番地が退避されているRAM402のアドレス4番地の下位8ビットをタイミングT2で、上位8ビットをタイミングT1でそれぞれABUSに出力するとともに、前サイクルのタイミングT1で読み出したタスク2のSPLを、タイミングT2でCBUSに出力する。また、タイミングT2ではコンテキスト退避メモリの33番地を読み出し、タイミングT1でCBUSに出力する。CBUSに出力されたSPLの内容およびSPHの内容はBBUSに転送され、レジスタセットBのSPLおよびSPHに格納される(ステップ54)。

【0081】<ステート4>CPUは、ALBに格納されたアドレスをIABに転送するとともに、AROMBUSを通してARAMBUSに出力し、RAM402の4番地からタスク2の戻り番地の下位8ビットを読み出して、LDBに格納する。

【0082】転送部は、前サイクルのタイミングT1でコンテキスト退避メモリの34番地から読み出したD0の内容を、CBUSに出力し、BBUSを通してレジスタセットBのD0に格納するとともに、タイミングT1でコンテキスト退避メモリの35番地のD1の内容を読み出してCBUSに出力し、BBUSを通してレジスタセットBのD1に格納する(ステップ54)。

【0083】<ステート5>CPUは、LDBの内容をABUSに出力し、ALUを通してALBに格納すると共に、前サイクルのIABの内容をインクリメンタにより+1したアドレスをARAMBUSに出力して、RAM401の5番地からタスク2の戻り番地の上位8ビットを読み出し、LDBに格納する(ステップ55)。

【0084】転送部は、ステート4と同様に、D2の内容およびD3の内容をコンテキスト退避メモリから読み出して、レジスタセットBのD2、D3にそれぞれ格納する(ステップ54)。

【0085】<ステート6>CPUは、ALBに格納されたタスク2の戻り番地の下位8ビットの内容をIABの下位8ビットに格納すると共に、LDBの内容をABUSに出力し、ALUを通して再度ABUSに出力する。CPUはABUSに出力したタスク2の戻り番地の上位8ビットをALUを通してALBに格納する。

【0086】転送部は、ステート4と同様に、A0の下位8ビットの内容および上位8ビットの内容をコンテキスト退避メモリから読み出して、レジスタセットBのA0L、A0Hにそれぞれ格納する(ステップ54)。

【0087】<ステート7>CPUは、ALBに格納されたタスク2の戻り番地の上位8ビットの内容をIABの上位8ビットに格納し(ステップ55)、IABの内容をAROMBUSに出力してタスク2の戻り番地から命令をフェッチしてIFBに格納する。

【0088】転送部は、ステート4と同様に、A1の下位8ビットの内容および上位8ビットの内容をコンテキスト退避メモリから読み出して、レジスタセットBのA1L、A1Hにそれぞれ格納する。

【0089】<ステート8>CPUは、BBUSに出力されたタスク2のPSWの内容をALUを通してABUSに出力し、PSW205に格納する(ステップ56)。また、前サイクルでフェッチした命令を命令バッファ206に転送すると共に、タスク2の戻り番地を+1したアドレスから続けて命令をフェッチし、IFBに格納する。

【0090】転送部は、前サイクルのタイミングT1でコンテキスト退避メモリの42番地から読み出したタスク2のPSWの内容を、タイミングT2でCBUSに出力すると共にBBUSに転送する。また、タイミングT2で読み出した43番地の内容を、タスクの実行時間を決めるタスクタイマ52に格納する(ステップ57)。

【0091】<ステート9>CPUは、タスク2の命令をIR207に格納して命令の解説を開始すると共に、BBUSに出力されたタスク2のOABの内容をALUを通してALBに格納する(ステップ58)。

【0092】転送部は、前サイクルのタイミングT1でコンテキスト退避メモリの44番地から読み出したタスク2のOABLの内容を、タイミングT2でCBUSに出力すると共にBBUSに転送する。また、タイミング

T2で読み出した45番地の内容を、タイミングT1でCBUSに出力すると共にBBUSに転送する。さらに、次タスク番号レジスタNEXTTSKの内容を現タスク番号レジスタCRNTTSKに転送し、NEXTTSKの内容を更新する。

【0093】転送部はさらに、レジスタセットA使用フラグ63を「0」にして、以降でCPUがタスクの実行に使用するレジスタセットをB側にする(ステップ59)。

【0094】<ステート10>CPUは、ALBに格納された内容をタスク2のOABに格納し、タスク2の命令を転送部とは独立して実行する(ステップ61)。従って、ステート11以降は、転送部のみの動作を説明する。

【0095】転送部は、タスク切り換え要求発生時のタスク1のコンテキストであるSPLおよびSPHをレジスタセットAから読み出してCBUSに出力し、コンテキスト退避メモリの16番地および17番地に格納する(ステップ62)。

【0096】<ステート11>転送部は、ステート10と同様にタスク切り換え要求発生時のタスク1のコンテキストであるD0およびD1をレジスタセットAから読み出してCBUSに出力し、コンテキスト退避メモリの18番地および19番地に格納する(ステップ62)。

【0097】<ステート12>転送部は、ステート10と同様にタスク切り換え要求発生時のタスク1のコンテキストであるD2およびD3をレジスタセットAから読み出してCBUSに出力し、コンテキスト退避メモリの20番地および21番地に格納する(ステップ62)。

【0098】<ステート13>転送部は、ステート10と同様にタスク切り換え要求発生時のタスク1のコンテキストであるA0LおよびA0HをレジスタセットAから読み出してCBUSに出力し、コンテキスト退避メモリの22番地および23番地に格納する(ステップ62)。

【0099】<ステート14>転送部は、ステート10と同様にタスク切り換え要求発生時のタスク1のコンテキストであるA1LおよびA1HをレジスタセットAから読み出してCBUSに出力し、コンテキスト退避メモリの24番地および25番地に格納する(ステップ62)。

【0100】図11は、タスク番号制御部62が次タスク番号レジスタNEXTTSKの値を決定するための動作を示した図である。

【0101】次タスク番号レジスタの値は、タスク許可フラグTSKEN56の内容と現タスク番号レジスタCRNTTSK60の内容とによって決定する。タスク許可フラグのLSB側が番号の若いタスクに対応しており、本実施例では、タスク許可フラグのLSB側が高い優先度となっている。従って、タスク許可フラグの複数

のビットがセットされている場合には、番号の若いタスクから先に実行される。例えば、TSKEN='0011'はタスク0とタスク1を許可することを示し、現タスク番号が「0」であれば次タスク番号は「1」になり、現タスク番号が「1」、「2」、「3」であれば次タスク番号は「0」になる。この例ではTSKENのビット2、ビット3は「0」になっているので、タスク2およびタスク3の許可フラグが命令の実行等でクリアされたことを示している。

【0102】他の組み合わせ状態も同様であるので、詳細な説明は省略する。なお、上記実施例の説明ではタスクの数を4にしたが、これに限定されるものではない。また、コンテキスト退避メモリやRAMへのタスクの退避領域のアドレスも限定されるものではない。

【0103】上記実施例では、タスクのコンテキストのうちプログラムカウンタのみをRAMに退避したが、プログラムカウンタ以外にPSW等をRAMに退避してもよい。

【0104】上記実施例では、RAMへの退避領域は特定番地としたが、実行中であったタスクのSPが指すスタック領域にコンテキストを退避してもよい。その場合、退避するSPはタスクの実行を停止した時点の値のままでもよい。

【0105】図12は、本発明の第2の実施例における転送部とCPUとの接続を説明する図である。

【0106】同図においてマイコンコア1は、CPU2とレジスタセット3とからなり、CPU2は、マイクロ制御部21と、コアレジスタ読み出し制御部22とバッファ23とを含む。拡張部4は、転送部5と2つのレジスタセットGRA6およびレジスタセットGRB7とからなり、転送部5は、拡張指定部51と拡張レジスタ読み出し制御部52とバッファ53とバッファ54とを含む。マイコンコア1と拡張部4とはABUSとBBUSからなる2本の内部バスで接続されている。マイコンコア1は、拡張部4が接続されていない状態でも命令の実行が可能である。以下では、マイコンコア1に拡張部4が接続されている場合の動作を説明する。

【0107】拡張指定部51により、拡張部4がマイコンコア1に接続されていることをCPU2に通知すると、コアレジスタ読み出し制御部22は、レジスタセットの読み出しをマイクロ制御部21から指示されても、バッファ23を動作させず、ABUSおよびBBUSには何も読み出さないようにする。一方、マイクロ制御部からのレジスタセット読み出しの指示は、拡張レジスタ読み出し制御部52に対しても出される。拡張部4に含まれる転送部5の動作は、2つのレジスタセットを有する本発明の第1の実施例と同様であるので詳細は省略する。

【0108】図13は、図12に示された本発明の第2の実施例における回路配置を説明する図である。

【0109】マイコンコア1はビット当りの回路の繰返しから構成されるデータバス部と不規則回路から構成される制御部とからなり、データバス部は制御部から出力されるデータバス制御信号により制御される。データバス部で処理するデータの流れの方向をX軸とすると、データバス制御信号はY軸方向である。制御部、データバス部、拡張部の順にY軸方向に配置し、データバス制御信号をデータバス部を貫通させてY軸方向に延長することで、同一のデータバス制御信号をデータバス部と拡張部とに接続する。

【0110】また、拡張部を制御する拡張部制御信号は、制御部から一旦X軸方向に出力した後、データバスを通らずに拡張部に接続する。

【0111】図13には省略しているが、ABUSおよびBBUSは、データバス部からY軸方向に延ばして拡張部に接続する。

【0112】なお、ABUSおよびBBUSは拡張部制御信号と同様に一旦X軸方向に延ばした後、拡張部に接続してもよい。

【0113】

【発明の効果】以上説明したように、請求項1の発明によれば、コンテキストの退避処理と切り換えた後のタスクの実行を並行処理できるという効果がある。特に、組み込み用途のマイクロコンピュータにおいては、ハードウェア規模の制約があっても、タスク切り換えを高速に実現できるので、リアルタイム制御を可能にする。

【0114】請求項2の発明によれば、請求項1の効果に加えて、コンテキストの退避処理を高速に実現できるという効果がある。

【0115】請求項3の発明によれば、請求項1または請求項2の効果に加えて、タスク切り換え要求が発生した後に、次に実行すべきタスクを決定できるという効果がある。これにより、リアルタイムOS等でタスク処理の順序を制御する応用においても柔軟に対応できる。

【0116】請求項4の発明によれば、請求項1の効果に加えて、コンテキストの退避処理と切り換えた後のタスクの実行との並行処理の時間を短縮する効果がある。これにより、並行処理中に命令によりコンテキスト退避メモリをアクセスするような場合、タスクの待ち時間が短縮できる。

【0117】請求項5の発明によれば、請求項1の効果に加えて、コンテキストを退避するメモリのアドレス計算時間を短縮するという効果がある。

【0118】請求項6の発明によれば、請求項1の効果に加えて、コンテキストの退避処理する時間を短縮するという効果がある。

【0119】請求項7の発明によれば、請求項6の効果に加えて、コンテキストの退避処理の量を減少するという効果がある。

【0120】請求項8の発明によれば、タスク切り換え

の機能をCPUに容易に追加して、機能拡張できるという効果がある。

【0121】請求項9の発明によれば、請求項8の効果に加えて、回路の配置をコンパクトにでき、チップの面積を小さくするという効果がある。

【0122】請求項10の発明によれば、タスク切り換えの順序をプログラムで柔軟に変更できるという効果がある。

【0123】請求項11の発明によれば、請求項10の効果に加えて、任意のタスクの起動をプログラムで柔軟に指定できるという効果がある。

【図面の簡単な説明】

【図1】本発明の実施例におけるマイクロコンピュータの主要な概略構成を示すブロック図

【図2】同実施例における転送部5の詳細な構成を示すブロック図

【図3】同実施例におけるマイクロコンピュータのより詳細な構成を示すブロック図

【図4】同実施例における転送制御部51のより詳細な構成と、コンテキスト退避メモリ3のより詳細なデータ構成を示すブロック図

【図5】同実施例におけるタスク切り換え時の処理のCPU2および転送部5の動作を示すフローチャート

【図6】同実施例におけるタスク切り換え時の処理のCPU2および転送部5の動作を示すタイミングチャート

【図7】同実施例におけるタスク切り換え時の処理のCPU2および転送部5の動作を示すタイミングチャート

【図8】同実施例におけるタスク切り換え時の処理のCPU2および転送部5の動作を示すタイミングチャート

【図9】同実施例における転送部5のシーケンサ55による各部の動作を示した図

【図10】同実施例における各タスクのプログラムカウンタを退避するRAM402の特定領域の構成を示した図

【図11】同実施例におけるタスク番号制御部62が、次タスク番号レジスタNEXTTSKの値を決定するための動作を示した図

【図12】本発明の第2の実施例におけるマイクロコンピュータの転送部とCPUとの接続を説明する図

【図13】同実施例における回路配置を説明する図

【図14】第1の従来技術におけるマルチタスク処理機能を有するマイクロコンピュータの説明図

【図15】第2の従来技術におけるマルチタスク処理機能を有するマイクロコンピュータの説明図

【符号の説明】

- 1 レジスタファイル
- 2 CPU
- 3 コンテキスト退避メモリ
- 4 メモリ
- 5 転送部



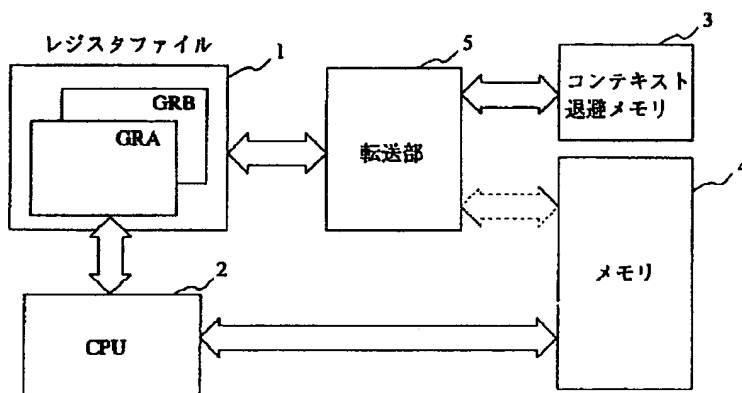
- ```

5 1 転送制御部
5 2 タスクタイマ
5 3 バススイッチ
5 4 RAMアドレス生成部
5 5 シーケンサ
5 6 タスク許可フラグ
5 7 次タスク番号レジスタ
5 8 コンテキスト退避メモリアドレスレジスタ
5 9 タスク許可フラグ制御部

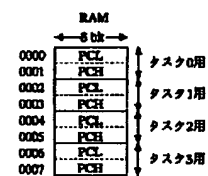
```

- |      |                    |
|------|--------------------|
| * 60 | 現タスク番号レジスタ         |
| 61   | 前タスク番号レジスタ         |
| 62   | タスク番号制御部           |
| 63   | レジスタセットA使用フラグ      |
| 64   | メモリ制御部             |
| 65   | コンテキスト退避メモリ書き込み制御部 |
| 66   | コンテキスト退避メモリ読み出し制御部 |
| 67   | レジスタ書き込み制御部        |
| 68   | レジスタ読み出し制御部        |

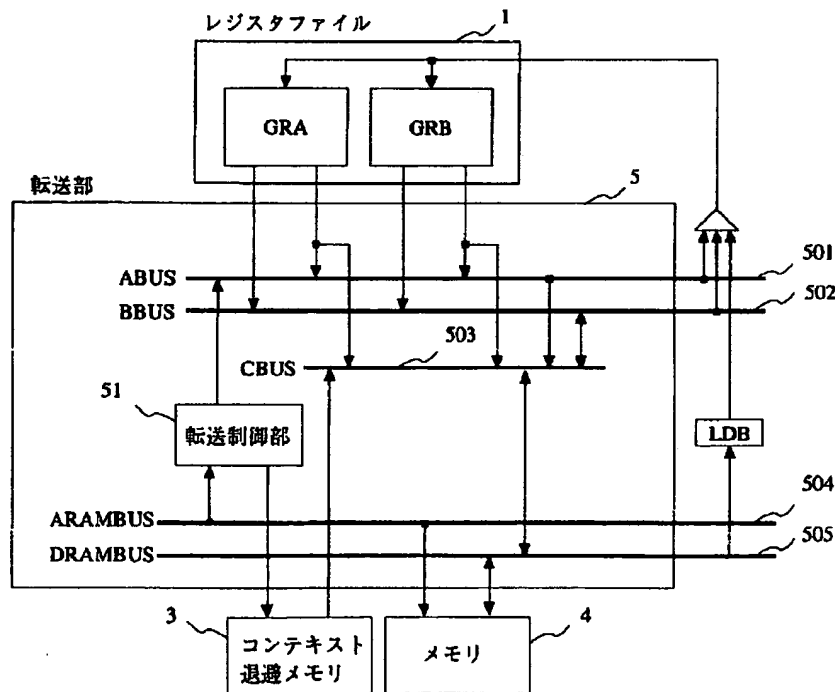
【圖 1】



【圖 10】

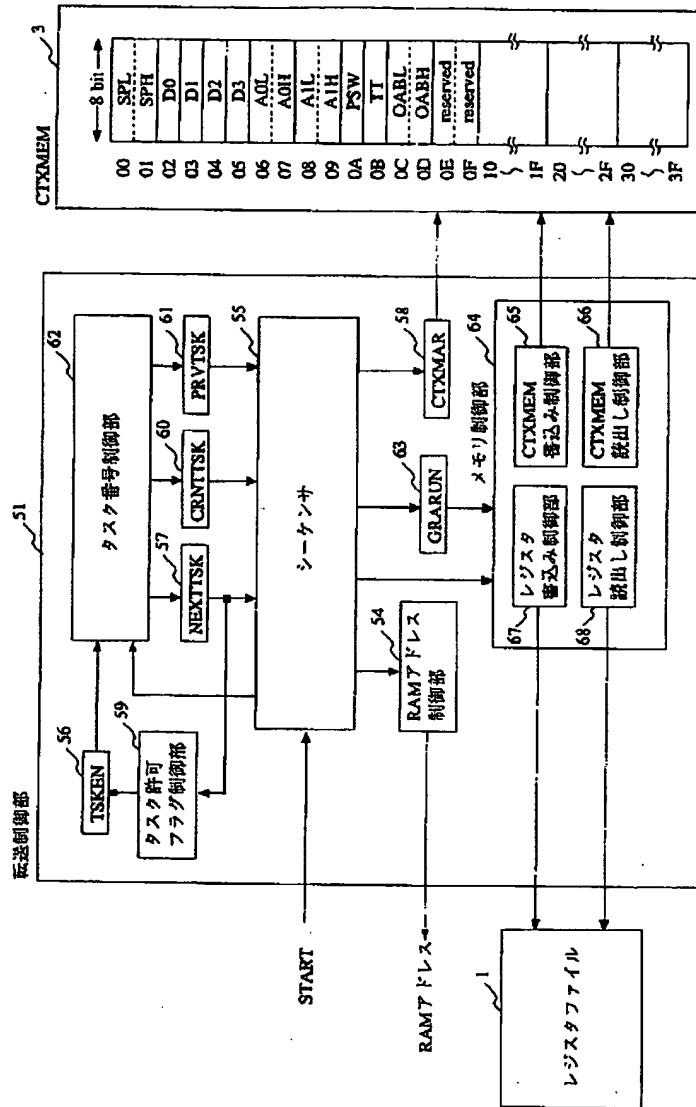


【図2】





【図4】



[illegible]

The timing diagram illustrates the execution of instructions across 16 cycles. The instructions and their corresponding operations are as follows:

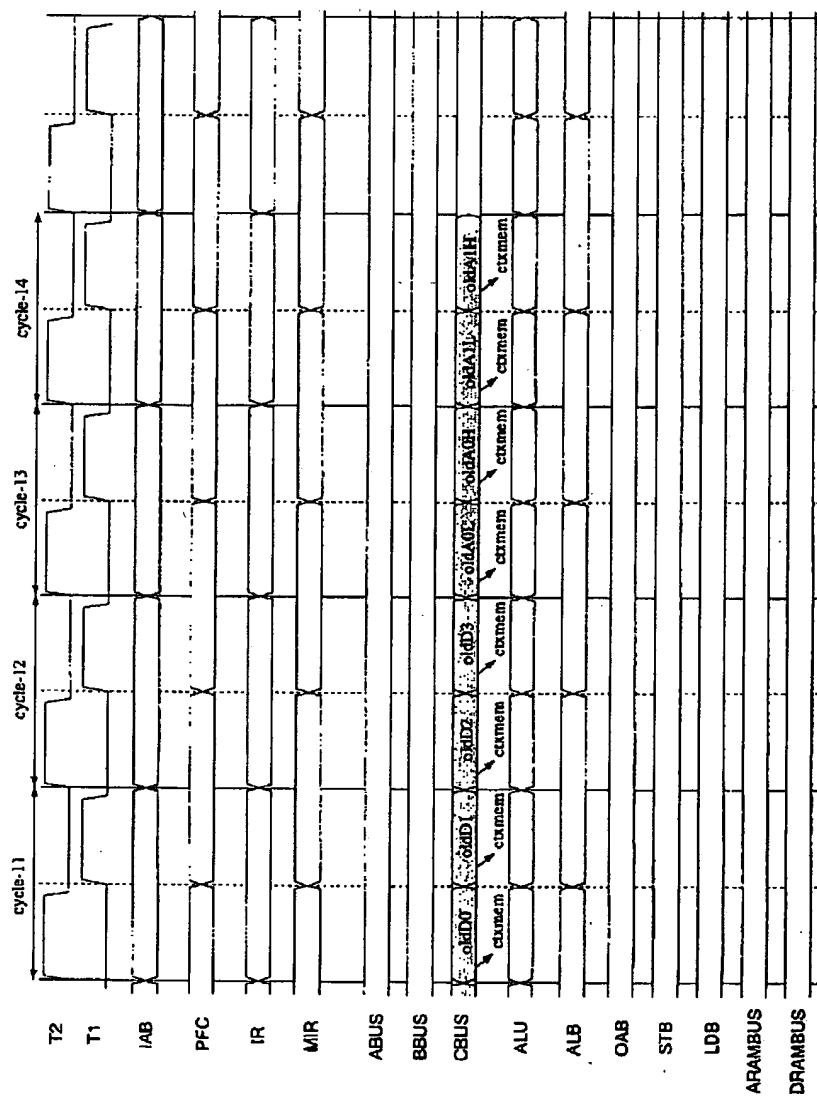
- T2**: Instruction execution.
- T1**: Instruction execution.
- LAB**: Instruction execution.
- PFC**: Instruction execution.
- IR**: Instruction execution.
- MIR**: Instruction execution.
- CRNTTSK\*16+12**: Instruction execution.
- oldPSW**: Instruction execution.
- oldPC-h**: Instruction execution.
- oldPC-l**: Instruction execution.
- PCsave\_addr+1**: Instruction execution.
- PCrestore\_addr**: Instruction execution.
- newPC-l**: Instruction execution.

The diagram also shows the state of various registers and flags during execution, including:

- Carry**: Indicated by a 'C' flag.
- Zero**: Indicated by a 'Z' flag.
- Overflow**: Indicated by a 'V' flag.
- Interrupt**: Indicated by an 'INT' flag.

[illegible]

【図8】



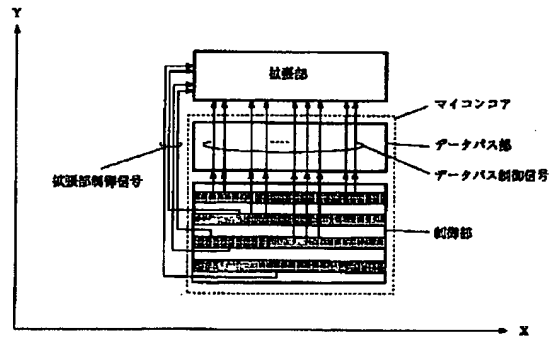
【図9】

| state | timing | CTXMEM    |         |     | BUS           |           |           | PRVTSK | CRNTTSK | NEXTTSK | GRARUN |
|-------|--------|-----------|---------|-----|---------------|-----------|-----------|--------|---------|---------|--------|
|       |        | address   | data    | R/W | ABUS          | BBUS      | CBUS      |        |         |         |        |
| 0     | T2     | crntsk+12 | oldOABL | W   | save_adr-l    | oldOABL   | (oldOABL) | 0      | 1       | 2       | 1      |
|       | T1     | crntsk+13 | oldOABH | W   | save_adr-h    | oldOABH   | (oldOABH) | 1      | 1       | 2       | 1      |
| 1     | T2     |           |         |     |               |           |           | 1      | 1       | 2       | 1      |
|       | T1     |           |         |     |               |           |           | 1      | 1       | 2       | 1      |
| 2     | T2     | crntsk+10 | oldPSW  | W   | oldPSW        |           |           | 1      | 1       | 2       | 1      |
|       | T1     | nextsk+0  | newSPL  | R   |               |           |           | 1      | 1       | 2       | 1      |
| 3     | T2     | nextsk+1  | newSPH  | R   | restore_adr-l | (newSPL)  | newSPL    | 1      | 1       | 2       | 1      |
|       | T1     | nextsk+2  | newD0   | R   | restore_adr-h | (newSPH)  | newSPH    | 1      | 1       | 2       | 1      |
| 4     | T2     | nextsk+3  | newD1   | R   |               | (newD0)   | newD0     | 1      | 1       | 2       | 1      |
|       | T1     | nextsk+4  | newD2   | R   |               | (newD1)   | newD1     | 1      | 1       | 2       | 1      |
| 5     | T2     | nextsk+5  | newD3   | R   |               | (newD2)   | newD2     | 1      | 1       | 2       | 1      |
|       | T1     | nextsk+6  | newA0L  | R   |               | (newD3)   | newD3     | 1      | 1       | 2       | 1      |
| 6     | T2     | nextsk+7  | newA0H  | R   |               | (newA0L)  | newA0L    | 1      | 1       | 2       | 1      |
|       | T1     | nextsk+8  | newA1L  | R   |               | (newA0H)  | newA0H    | 1      | 1       | 2       | 1      |
| 7     | T2     | nextsk+9  | newA1H  | R   |               | (newA1L)  | newA1L    | 1      | 1       | 2       | 1      |
|       | T1     | nextsk+10 | newPSW  | R   |               | (newA1H)  | newA1H    | 1      | 1       | 2       | 1      |
| 8     | T2     | nextsk+11 | newTT   | R   |               | (newPSW)  | newPSW    | 1      | 1       | 2       | 1      |
|       | T1     | nextsk+12 | newOABL | R   |               | newTT     | newTT     | 1      | 1       | 2       | 1      |
| 9     | T2     | nextsk+13 | newOABH | R   |               | (newOABL) | newOABL   | 1      | 1       | 2       | 0      |
|       | T1     |           |         |     |               | (newOABH) | newOABH   | 1      | 2       | 3       | 0      |
| 10    | T2     | prvtsk+0  | oldSPL  | W   |               |           | oldSPL    | 1      | 2       | 3       | 0      |
|       | T1     | prvtsk+1  | oldSPH  | W   |               |           | oldSPH    | 1      | 2       | 3       | 0      |
| 11    | T2     | prvtsk+2  | oldD0   | W   |               |           | oldD0     | 1      | 2       | 3       | 0      |
|       | T1     | prvtsk+3  | oldD1   | W   |               |           | oldD1     | 1      | 2       | 3       | 0      |
| 12    | T2     | prvtsk+4  | oldD2   | W   |               |           | oldD2     | 1      | 2       | 3       | 0      |
|       | T1     | prvtsk+5  | oldD3   | W   |               |           | oldD3     | 1      | 2       | 3       | 0      |
| 13    | T2     | prvtsk+6  | oldA0L  | W   |               |           | oldA0L    | 1      | 2       | 3       | 0      |
|       | T1     | prvtsk+7  | oldA0H  | W   |               |           | oldA0H    | 1      | 2       | 3       | 0      |
| 14    | T2     | prvtsk+8  | oldA1L  | W   |               |           | oldA1L    | 1      | 2       | 3       | 0      |
|       | T1     | prvtsk+9  | oldA1H  | W   |               |           | oldA1H    | 1      | 2       | 3       | 0      |

【図11】

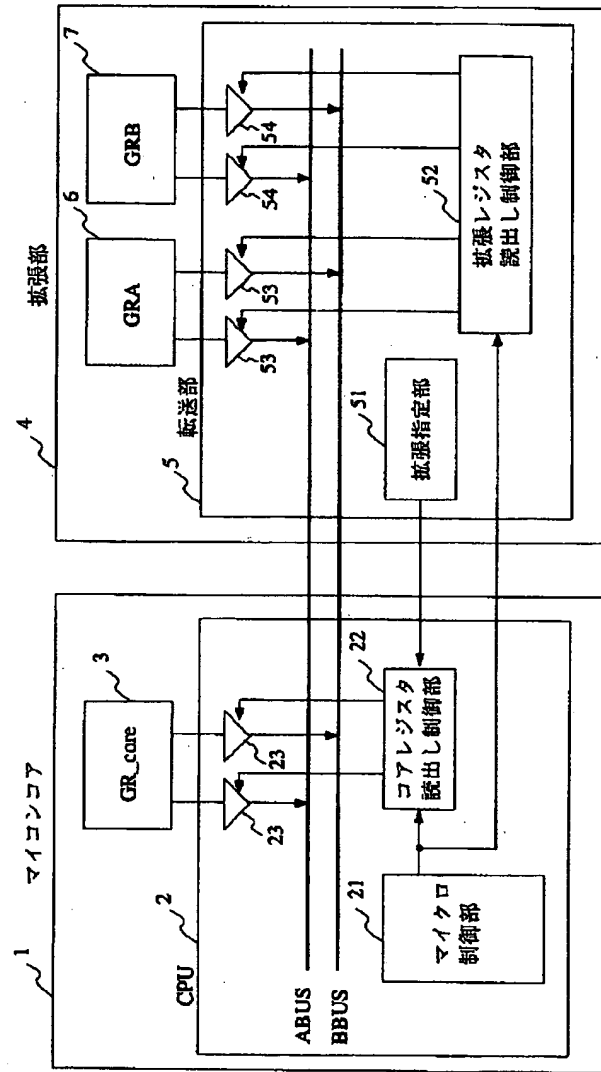
| TSKPN | CRNTTSK | NEXTTSK |
|-------|---------|---------|
| 000   | 0       | 0       |
| 0010  | 1       | 1       |
| 0011  | 0       | 1       |
|       | 1       | 0       |
|       | 2       | 0       |
|       | 3       | 0       |
| 0100  | 0       | 2       |
| 0101  | 1       | 2       |
|       | 2       | 0       |
|       | 3       | 0       |
| 0110  | 0       | 2       |
|       | 1       | 2       |
|       | 2       | 1       |
| 0111  | 0       | 1       |
|       | 1       | 2       |
|       | 2       | 0       |
|       | 3       | 0       |
| 1000  | 0       | 3       |
| 1001  | 1       | 3       |
|       | 2       | 3       |
|       | 3       | 0       |
| 1010  | 0       | 1       |
|       | 1       | 3       |
|       | 2       | 3       |
|       | 3       | 1       |

【図13】

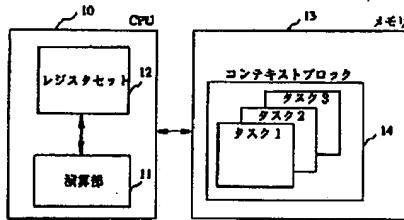




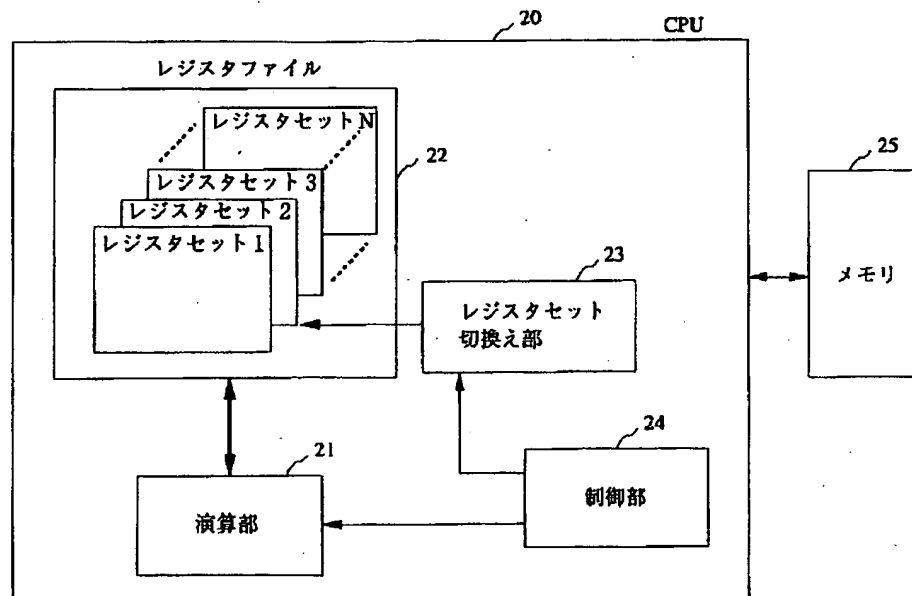
【図12】



【図14】



【図15】



**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☒ **BLACK BORDERS**
- ☒ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☒ **FADED TEXT OR DRAWING**
- ☒ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☒ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☒ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**